LISTING OF CLAIMS

The listing of claims provided below replaces all prior versions, and listings, of claims in the application.

1. (Currently Amended) A method for obtaining traces of a program, comprising:

(a) obtaining an original set of instructions which define the program, wherein the original set of instructions does not include an instrumentation instruction;

(b) obtaining an instrumented version of the original set of instructions, wherein the instrumented version of the original set of instructions includes each instruction in the original set of instructions and a number of instrumentation instructions defined to generate traces, wherein the number of instrumentation instructions are dispersed in a substantially uniform manner throughout the instrumented version of the original set of instructions;

(c) executing the [[~~an~~]] original set of instructions;

(d) switching execution from the original set of instructions to the [[~~an~~]] instrumented version of the original set of instructions upon encountering a first trigger condition; [[~~and~~]]

(e) executing the instrumented version of the original set of instructions so as to generate ~~generating~~ traces through execution of one or more of the number of instrumentation instructions ~~contained within the instrumented version of the original set of instructions~~;

(f) switching execution from the instrumented version of the original set of instructions back to the original set of instructions upon encountering a second trigger condition; and

(g) repeating operations (c) through (f).

2. (Original) A method for obtaining traces of a program as recited in claim 1, wherein the switching of execution from the original set of instructions to the instrumented version of the original set of instructions occurs at a location of known state in the original set of instructions.

3. (Currently Amended) A method for obtaining traces of a program as recited in claim 1, ~~further comprising:~~

~~triggering the switching of execution from the original set of instructions to the instrumented version of the original set of instructions,~~ wherein the first trigger condition ~~triggering~~ is based on an elapsed time of execution, wherein encountering the first trigger condition causes ~~triggering causing~~ the switching of execution from the original set of instructions to the instrumented version of the original set of instructions to occur at a next location of known state in the original set of instructions.

4. (Currently Amended) A method for obtaining traces of a program as recited in claim 3, wherein the first trigger condition ~~triggering~~ is defined ~~performed~~ such that execution of the original set of instructions accounts for more than about 90 percent of the elapsed time of execution.

5. (Currently Amended) A method for obtaining traces of a program as recited in claim 1, ~~further comprising:~~

~~triggering a switching of execution from the instrumented version of the original set of instructions back to the original set of instructions,~~ wherein the second trigger

condition ~~triggering~~ is based on an elapsed time of execution, wherein encountering the second trigger condition causes ~~triggering causing~~ the switching of execution from the instrumented version of the original set of instructions back to the original set of instructions to occur at a next location of known state in the instrumented version of the original set of instructions~~; and~~

~~switching execution from the instrumented version of the original set of instructions back to the original set of instructions~~.

6. (Original) A method for obtaining traces of a program as recited in claim 5, wherein the next location of known state in the instrumented version of the original set of instructions corresponds to an instruction common to both the instrumented version of the original set of instructions and the original set of instructions.

7. (Currently Amended) A method for obtaining traces of a program as recited in claim 5, wherein the second trigger condition ~~triggering~~ is defined ~~performed~~ such that execution of the instrumented version of the original set of instructions accounts for less than about 10 percent of the elapsed time of execution.

8. (Original) A method for obtaining traces of a program as recited in claim 1, wherein execution of the instrumented version of the original set of instructions is performed by an emulator.

9. (Currently Amended) A method for obtaining traces of a program, comprising:

(a) executing an original code which defines the program, wherein the original code does not include an instrumentation instruction;

(b) switching execution from the original code to an instrumented code, wherein the instrumented code includes each instruction present in the original code and a number of instrumentation instructions dispersed in a substantially uniform manner throughout the instrumented code, wherein the number of instrumentation instructions are defined to generate traces;

(c) executing the instrumented code so as to generate traces through execution of one or more of the instrumentation instructions;

~~generating traces; and~~

(d) switching execution from the instrumented code back to the original code; and

(e) repeating operations (a) through (d).

10. (Original) A method for obtaining traces of a program as recited in claim 9, further comprising:

triggering the switching of execution from the original code to the instrumented code, the triggering causing the switching of execution to occur at a next location of known state in the original code.

11. (Original) A method for obtaining traces of a program as recited in claim 10, wherein the triggering is based on an elapsed time of execution, the triggering being performed such that execution of the original code accounts for more than about 90 percent of the elapsed time of execution.

12. (Currently Amended) A method for obtaining traces of a program as recited in claim 9, further comprising:

triggering the switching of execution from the instrumented code back to the original code, the triggering causing the switching of execution to occur at a next location of known state in the instrumented code.

13. (Original) A method for obtaining traces of a program as recited in claim 12, wherein the next location of known state in the instrumented code corresponds to an instruction common to both the instrumented code and the original code.

14. (Original) A method for obtaining traces of a program as recited in claim 12, wherein the triggering is based on an elapsed time of execution, the triggering being performed such that execution of the instrumented code accounts for less than about 10 percent of the elapsed time of execution.

15. (Currently Amended) A method for obtaining traces of a program as recited in claim 9, wherein both switching execution from the original code to the instrumented code and switching execution from the instrumented code back to the original code are performed using return addresses during processing of function calls.

16. (Original) A method for obtaining traces of a program as recited in claim 9, further comprising:

defining a map of instruction addresses, the map of instruction addresses identifying correspondences between instruction addresses in the original code and instruction addresses in the instrumented code.

17. (Currently Amended) A method for obtaining traces of a program as recited in claim 16, wherein both switching execution from the original code to the instrumented code and switching execution from the instrumented code back to the original code are performed using the map of instruction addresses.

18. (Currently Amended) A computer readable medium ~~media~~ containing program instructions for obtaining traces of a program, comprising:

program instructions for executing an original code, wherein the original code does not include an instrumentation instruction;

program instructions for switching execution from the original code to an instrumented code, wherein the instrumented code includes each instruction present in the original code and a number of instrumentation instructions dispersed in a substantially uniform manner throughout the instrumented code, wherein the number of instrumentation instructions are defined to generate traces;

program instructions for executing the instrumented code so as to generate traces through execution of one or more of the instrumentation instructions;

~~program instructions for generating traces;~~ and

program instructions for switching execution from the instrumented code back to the original code.

19. (Currently Amended) A computer readable medium ~~media~~ containing program instructions for obtaining traces of a program as recited in claim 18, further comprising:

program instructions for triggering a switching of execution from the original code to the instrumented code, the program instructions for triggering causing the switching of execution to occur at a next location of known state in the original code.

20. (Currently Amended) A computer readable medium ~~media~~ containing program instructions for obtaining traces of a program as recited in claim 19, wherein the program instructions for triggering are based on an elapsed time of execution, the program instructions for triggering causing execution of the original code to account for more than about 90 percent of the elapsed time of execution.

21. (Currently Amended) A computer readable medium ~~media~~ containing program instructions for obtaining traces of a program as recited in claim 18, further comprising:

program instructions for triggering a switching of execution from the instrumented code back to the original code, the program instructions for triggering causing the switching of execution to occur at a next location of known state in the instrumented code.

22. (Currently Amended) A computer readable medium ~~media~~ containing program instructions for obtaining traces of a program as recited in claim 21, wherein the next location of known state in the instrumented code corresponds to an instruction common to both the instrumented code and the original code.

23. (Currently Amended) A computer readable medium ~~media~~ containing program instructions for obtaining traces of a program as recited in claim 21, wherein the

program instructions for triggering are based on an elapsed time of execution, the program instructions for triggering causing execution of the instrumented code to account for less than about 10 percent of the elapsed time of execution.

24. (Currently Amended) A computer readable medium ~~media~~ containing program instructions for obtaining traces of a program as recited in claim 18, wherein the program instructions for switching execution from the original code to the instrumented code and the program instructions for switching execution from the instrumented code back to the original code are defined to use return addresses during processing of function calls to effect the switching.

25. (Currently Amended) A computer readable medium ~~media~~ containing program instructions for obtaining traces of a program as recited in claim 18, further comprising:

program instructions for defining a map of instruction addresses, the map of instruction addresses identifying correspondences between instruction addresses in the original code and instruction addresses in the instrumented code.

26. (Currently Amended) A computer readable medium ~~media~~ containing program instructions for obtaining traces of a program as recited in claim 25, wherein the program instructions for switching execution from the original code to the instrumented code and the program instructions for switching execution from the instrumented code back to the original code are defined to use the map of instruction addresses to effect the switching.